



# EXPERIMENT - 13

## Object Oriented Programming Lab

### Aim

Write a program to enter any number and find its factorial using constructor.

Syeda Reeha Quasar

14114802719

4C7

## EXPERIMENT – 13

### Aim:

Write a program to enter any number and find its factorial using constructor.

### Source Code:

```
#include<iostream>
using namespace std;

class factorial{
    int n, i, f;

public:
    factorial(){
        cout << "Enter a number to calculate factorial of: ";
        cin>>n;

        f = 1;
        for (i = 1; i <= n; i++) {
            f = f * i;
        }
    }

    void executeFac() {
        cout << "Factorial of " << n << " is: " << f;
    }
};

int main() {

    factorial factObj1;
    factObj1.executeFac();

    return 0;
}
```

## Output:

```
PS D:\sem 4\cpp\oops> cd "d:\sem 4\cpp\oops\" ; if ($?) { g++ factorialUsingConstructor.cpp -o factorialUsingConstructor } ; if ($?) {  
.\factorialUsingConstructor }  
Enter a number to calculate factorial of: 6  
Factorial of 6 is: 720
```

```
PS D:\sem 4\cpp\oops> .\factorialUsingConstructor  
Enter a number to calculate factorial of: 10  
Factorial of 10 is: 3628800
```

## Viva Questions

### *Q1) When are copy constructors called in C++?*

There are some possible situation when copy constructor called in C++,

- When an object of the class is returned by value.
- When an object of the class is passed (to a function) by value as an argument.
- When an object is constructed based on another object of the same class.
- When the compiler generates a temporary object.

### *Q2) Why copy constructor takes the parameter as a reference in C++?*

A copy constructor is called when an object is passed by value. The copy constructor itself is a function. So if we pass an argument by value in a copy constructor, a call to copy constructor would be made to call copy constructor which becomes a non-terminating chain of calls. Therefore compiler doesn't allow parameters to be passed by value.

### *Q3) Why copy constructor argument should be const in C++?*

There are some important reasons to use const in the copy constructor.

- const keyword avoids accidental changes.
- You would like to be able to create a copy of the const objects. But if you're not passing your argument with a const qualifier, then you can't create copies of const objects.
- You couldn't create copies from temporary reference, because temporary objects are rvalue, and can't be bound to reference to non-const.